

(19)



**Евразийское
патентное
ведомство**

(11) **040377**(13) **B1**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2022.05.25

(21) Номер заявки
201990407

(22) Дата подачи заявки
2019.02.27

(51) Int. Cl. **G06F 17/00** (2019.01)
G06N 3/00 (2006.01)

(54) СПОСОБ ОТЛАДКИ ОБУЧЕННОЙ РЕКУРРЕНТНОЙ НЕЙРОННОЙ СЕТИ

(31) **2019103863**

(32) **2019.02.12**

(33) **RU**

(43) **2020.08.31**

(71)(73) Заявитель и патентовладелец:
**ПУБЛИЧНОЕ АКЦИОНЕРНОЕ
ОБЩЕСТВО "СБЕРБАНК
РОССИИ" (ПАО СБЕРБАНК) (RU)**

(72) Изобретатель:
**Жаров Ярослав Максимович,
Корженков Денис Михайлович (RU)**

(74) Представитель:
Герасин Б.В. (RU)

(56) Long Short-Term Memory-Networks for Machine -
<https://arxiv.org/pdf/1601.06733.pdf>
Deep Learning, NLP, and Representations Posted on
July 7, 2014 - <http://colah.github.io/posts/2014-07-NLP-RN-Ns-Representations/>
Understanding LSTM Networks Posted on August
27, 2015 - <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
Fine-grained Opinion Mining with Recurrent Neural
Networks and Word Embeddings - <https://www.aclweb.org/anthology/D15-1168>

The Secret Sharer: Evaluating and
Testing Unintended Memorization in Neural
Networks - <https://arxiv.org/pdf/1802.08232.pdf?cmp=em-data-na-na-newsltr20180321&imm=0fc19d>

Neural Networks, Manifolds, and Topology Posted
on April 6, 2014 - <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>

Attention and Augmented Recurrent Neural
Networks - <https://distill.pub/2016/augmented-rnns/>

TYING WORD VECTORS AND WORD
CLASSIFIERS : A LOSS FRAMEWORK FOR
LANGUAGE MODELING <https://arxiv.org/pdf/1611.01462.pdf>

A Critical Review of Recurrent Neural Networks for
Sequence Learning <https://arxiv.org/pdf/1506.00019.pdf>

Sequence-to-Sequence Learning as Beam-Search
Optimization <https://arxiv.org/pdf/1606.02960.pdf>

Automatic speech recognition for under-resourced
languages: A survey <https://repository.nwu.ac.za/bitstream/handle/10394/26495/barnard-2014-Automatic-speech-recognition-for-under-resourced-languages-1>

Learning Phrase Representations using RNN
Encoder-Decoder for Statistical Machine Translation -
<https://arxiv.org/pdf/1406.1078v3.pdf>

NEURAL MACHINE TRANSLATION BY
JOINTLY LEARNING TO ALIGN AND TRANSLATE -
<https://arxiv.org/pdf/1409.0473.pdf>

Overcoming the Curse of Sentence Length for
Neural Machine Translation using Automatic Segmentation -
<https://www.aclweb.org/anthology/W14-4009>

US-B2-9165254
US-B1-9015093

(57) Настоящее решение относится к области обработки данных, в частности к способу для работы с обученными нейронными сетями (НС) и их отладки. Достижимый технический эффект от применения заявленного способа заключается в обеспечении возможности оценить влияние входных возмущений на результат вычисления агрегирующей функции от скрытых состояний РНС, за счет минимизации меры расхождения при поиске релевантных подпоследовательностей токенов. Указанный технический результат достигается за счет компьютерно-реализуемого способа отладки обученной рекуррентной нейронной сети (РНС), выполняемого с помощью по меньшей мере одного процессора, причем РНС обучена на наборе данных, состоящем из последовательностей токенов, которые являются векторными представлениями элементов упомянутого набора данных, и способ содержит этапы, на которых: а) получают значение агрегирующей функции скрытых состояний РНС для упомянутой последовательности токенов; б) осуществляют поиск внутри упомянутой последовательности токенов по меньшей мере одной подпоследовательности токенов и определяют для каждой упомянутой подпоследовательности агрегирующую функцию скрытых состояний РНС; в) определяют подпоследовательность токенов на основании минимального

B1**040377****040377 B1**

значения меры расхождения между значениями агрегирующих функций, полученных на этапах а) и б).

040377 B1

040377 B1

Область техники

Настоящее техническое решение в общем относится к области обработки данных, в частности к способу для работы с обученными искусственными нейронными сетями (далее - ИНС) и их отладки.

Уровень техники

Проблематика в работе по интерпретации/отладке ИНС на данный момент все более набирает вес для систем, использующих решения на базе моделей машинного обучения, из-за отсутствия понимания экспертами работы ИНС. Тем не менее, существующие методы интерпретации/отладки, пригодные для работы с полностью связными и сверточными архитектурами ИНС, часто не могут напрямую применяться для рекуррентных архитектур ИНС, которые широко применяются в решении задач по обработке естественного языка (англ. NLP - Natural Language Processing), обработке медицинских данных (англ. EHR - Electronic Health Records), прогнозированию временных рядов и других сфер. Проблема чаще всего связана напрямую с вариативностью длины входной последовательности данных и особенностей их строения.

Работы по созданию последовательных архитектур, интерпретируемых по построению, предлагались в источниках информации Choi et al. [1], [2], Ma et al. [3]. Также, в данной области работы и применения ИНС велись исследования по изменению значений скрытых состояний, получаемых при работе ИНС, для поиска соответствующих паттернов входной информации (Karpathy et al. [4], Hasani et al. [5], Strobel et al. [6]).

Рекуррентные искусственные нейронные сети (Recurrent Neural Network, RNN или РНС) - класс моделей машинного обучения, основанный на использовании предыдущих состояний сети для вычисления текущего. Такие сети удобно применять в тех случаях, когда входные данные задачи представляют собой последовательность значений - токенов, как, например, текстовые данные, где текстовый фрагмент представлен нефиксированным количеством предложений, фраз и слов. Токенами являются векторные представления элементов набора данных, используемого для обучения ИНС. Каждый символ в тексте, отдельные слова, знаки препинания и даже целые фразы - все это может являться атомарным элементом входной последовательности.

Некоторые работы осуществлялись для открытия свойств конкретных рекуррентных блоков - LSTM (англ. Long short-term memory/Долгая краткосрочная память). В одной из работ авторов Murdoch & Szlam [7] предлагается выражать выходное состояние последнего скрытого состояния LSTM как сумму вкладов токенов и впоследствии строить классификатор на основе набора правил, приближающий исходную РНС. Однако, данный способ является достаточно ненаправленным, поскольку смешивает состояния ячеек от всех временных меток и выходное значение последнего шага.

В работе Murdoch et al. [8] представлена контекстная аддитивная декомпозиция выхода LSTM, которая захватывает вклад комбинаций входных токенов. Такая декомпозиция использует линеаризацию гиперболического тангенса и сигмоидальной функции - таким образом, несмотря на сохранение оригинальной рекуррентной ячейки неизменной, подход не может считаться исчерпывающим.

Тем не менее, такие подходы имеют ограниченную сферу применения, в частности, они являются малоприменимыми для обработки данных финансовых транзакций или логов, получаемых от устройств Интернета Вещей (англ. аббр. - IoT), что связано со сложностью семантического распознавания значения паттернов в таких типах данных.

Раскрытие изобретения

Для решения существующей технической проблемы или технической задачи в данной области предлагается рассмотреть РНС ячейку в процессе фазы ее работы при вынесении заключения (суждения), выполняемого РНС, как динамическую систему (S, s_0) , которая обновляет ее состояние $s_{t-1} \in \mathbb{R}^h$ в момент времени t под влиянием внешних возмущений $x_t \in \mathbb{R}^d$, согласно формуле

$$s_t = S(s_{t-1}, x_t), \quad (1)$$

$$S : \mathbb{R}^h \times \mathbb{R}^d \rightarrow \mathbb{R}^h$$

Например, если рассмотреть частный вариант для решения задачи классификации анализ тональности текста, то такая система инициализируется со стартового состояния s_0 , которое замораживается после стадии обучения (обычно с нулевым значением). Затем система обновляет свое состояние при обработке векторных представлений слов (токенов)

$$X = \{x_i\}_{i=1}^T$$

документа. Наконец, последнее состояние системы передается полностью связному классификационному слою. Данный пример представлен для текстового классификатора, однако применение настоящего решения не ограничивается исключительно данной областью и может использоваться для различных задач классификации по различным типам входных данных, например данных банковских транзакций, телеметрической информации и др.

Для решения задачи снижения вычислительной нагрузки на стадии обучения и работы РНС выполнялись различные исследования, в частности, представленные в приведенных далее источниках информации. Одним из примеров является метод LSTM-Jump, предложенный в работе авторов Yu et al. [9], который прогнозирует количество токенов, которые необходимо пропустить, с применением обучающего алгоритма с подкреплением. В некоторых примерах также решается проблема ранней остановки -

получения предсказания РНС, не доводя обработку входной последовательности токенов до конца (см. источник информации Ryabinin & Lobacheva [10]).

Таким образом, можно сделать вывод о том, что многие внешние возмущения не изменяют значительно результирующее состояние системы и могут быть исключены без каких-либо негативных последствий для качества модели машинного обучения. Исходя из данного вывода можно представить выраженные состояния РНС ячейки s_t в следующем виде:

$$s_t = u_t \cdot S(s_{t-1}, x_t) + (1 - u_t) \cdot s_{t-1}, \quad (2)$$

где:

$u_t \in \{0, 1\}$ и указывает, использовалось ли то или иное векторное представление слова x_t в момент времени t для обновления состояния системы или нет;

$U = (u_1, \dots, u_t)$ - вектор, содержащий бинарные переменные для всей последовательности нарушений X . Для стандартной РНС

$$U = (1, 1, \dots, 1) = \vec{1}.$$

В качестве примера можно принять

$$s_T^{(U)}$$

как конечное состояние системы, обновленное путем применения формулы (2) с заданным вектором U . Предположим, что

$$D(s; \tilde{s})$$

представляет собой меру расхождения между двумя состояниями системы. Таким образом, для решения поставленной задачи необходима минимизация величины

$$D(s_T^{(\vec{1})}, s_T^{(U)})$$

с одновременным уменьшением суммы элементов (токенов) U .

Достижимый технический эффект от применения заявленного способа заключается в обеспечении возможности оценить влияние входных возмущений на результат вычисления агрегирующей функции от скрытых состояний РНС, за счет минимизации меры расхождения при поиске релевантных подпоследовательностей токенов.

Указанный технический результат достигается за счет осуществления компьютерно-реализуемого способа отладки обученной рекуррентной нейронной сети (РНС), выполняемый с помощью по меньшей мере одного процессора, причем РНС обучена на наборе данных, состоящем из последовательностей токенов, которые являются векторными представлениями элементов упомянутого набора данных, и способ содержит этапы, на которых:

а) получают значение агрегирующей функции скрытых состояний РНС для упомянутой последовательности токенов;

б) осуществляют поиск внутри упомянутой последовательности токенов по меньшей мере одной подпоследовательности токенов и определяют для каждой упомянутой подпоследовательности агрегирующую функцию скрытых состояний РНС;

с) определяют подпоследовательность токенов на основании минимального значения меры расхождения между значениями агрегирующих функций, полученных на этапах а) и б).

В частном варианте осуществления способа мера расхождения является заданной функцией.

В другом частном варианте осуществления способа мера расхождения является обучаемой моделью машинного обучения, представляющей собой по меньшей мере одну искусственную нейронную сеть, принимающую на вход значения агрегирующих функций.

Краткое описание чертежей к описанию

Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания и прилагаемых чертежей.

Фиг. 1 иллюстрирует пример исходной последовательности.

Фиг. 2 иллюстрирует пример выявленной подпоследовательности.

Фиг. 3 иллюстрирует блок-схему выполняемого способа отладки РНС.

Фиг. 4-7 иллюстрируют примеры работы отладки РНС с помощью предложенного метода. На них показаны вероятности включения каждого токена последовательности в искомую подпоследовательность.

Фиг. 8 представляет пример вычислительной системы для реализации способа.

Подробное описание технического решения

Для реализации заявленного способа предлагается осуществлять анализ исходной последовательности токенов (10), представленной на фиг. 1, которая использовалась для обучения РНС на предмет выявления такой подпоследовательности токенов (11) (фиг. 2), которая будет давать минимальное значение степени расхождения D , что позволит выполнить отладку/интерпретацию РНС.

На фиг. 3 представлен пример выполнения этапов при реализации заявленного способа отладки РНС. РНС в конкретном примере реализации были обучены на наборе данных, включающем последовательности (10) текстовых данных - рецензий, представленных в виде токенов. В данном случае токен представляет собой векторное представление каждого слова из последовательности (10). Векторное представление слов может создаваться с помощью таких алгоритмов, как: Word2Vec, Glove, OneHotEncoding и др.

Для исходной последовательности токенов (10), с помощью которой была обучена РНС (201), определяется значение агрегирующей функции скрытых состояний РНС (202). Например, в качестве агрегирующей функции для последовательности (10) может использоваться выбор последнего элемента последовательности, т.е. последнее из всех состояний s системы.

Для обучающей последовательности (10) токенов осуществляется поиск и выявление подпоследовательности (11) токенов (203), для которой выполняется определение агрегирующей функции (204). Сущность заявленного технического решения состоит в том, чтобы выявить подпоследовательность токенов (11) в каждой исходной последовательности (10) данного набора данных таким образом, чтобы значение агрегирующей функции от скрытых состояний, полученных при прохождении данной РНС по выбранной на этапе (204) подпоследовательности (11), отличалось как можно меньше от значения агрегирующей функции от скрытых состояний, полученных при прохождении данной РНС по исходной последовательности (10), что проверяется для каждой выявленной подпоследовательности на этапе (205).

На фиг. 3-6 представлены примеры выполненных работ по отладке/интерпретации РНС. В данных примерах РНС являлась частью системы по классификации текстовой информации, в частности, рецензий пользователей о кинофильмах.

Подпоследовательность токенов (11) может содержать последовательные токены, так и токены, удаленные друг от друга. В каждой последовательности (10) анализируется несколько подпоследовательностей (11), из которых выбирается подпоследовательность (11) с минимальным значением меры расхождения D (206) между значениями агрегирующих функций исходной последовательности (10) и найденной подпоследовательности (11).

В качестве меры расхождения

$$D(s; \tilde{s})$$

значений агрегирующей функции могут использоваться, например, евклидово расстояние (иногда называют метрикой), относительное евклидово расстояние, косинусное расстояние и т.д. Также эти метрики могут применяться как к самим значениям агрегирующей функции, так и к неким производным от нее значениям. Например, представления, полученные полносвязными слоями, которые на основании входных данных, обработанных РНС, подсчитывают N -мерный вектор, где N - число классов.

Также, построение обучаемой метрики расхождения может выполняться подобно тому, как в генеративных конкурирующих сетях (GAN) [11] строится дискриминатор, т.е. с использованием отдельной нейронной сети для подсчета расстояния.

При нахождении требуемой подпоследовательности (11), выявляются токены, которые оказались важны для данной РНС. Например, если в классификаторе рецензий, как представлено на фиг. 1 - фиг. 2, на фильмы РНС обрабатывает слова "замечательно" и "плохой", то РНС работает ожидаемым образом, поскольку выявляются слова, непосредственно относящиеся к задаче РНС. В случае же, если РНС обрабатывает слова "ты", "мы" и т.п., т.е. токены, не относящиеся явным образом к предмету классификации, то такая РНС, с высокой вероятностью переобучилась и требует коррекции. Под переобучением в данном случае понимается способность к правильной классификации исходя из особенностей конкретного набора данных, вместо обобщения понятий хороших и плохих отзывов. Используя принцип выявления подпоследовательности (11) для анализа работы РНС, можно значительно сократить время анализа состояния РНС, и упростить анализ ошибок в ее работе.

В одном из частных примеров реализации заявленного способа предпочтительно выбирать подпоследовательность также наименьшей длины, исходя из количества токенов. Например, это может быть достигнуто за счет включения в функцию потерь компоненты, отвечающей за количество выбранных токенов. Данный пример осуществления может быть реализован с помощью любого алгоритма оптимизации функции потерь, например, с помощью градиентного спуска.

Далее рассмотрим частные примеры выявления подпоследовательностей. В одном случае каждому токеноу в исходной последовательности (10) сопоставляется метка, которая может принимать два состояния - 0 или 1, характеризующая включение данного токена в подпоследовательность (11), причем данный этап выполняется дифференцируемым образом (например, с помощью straight-through estimator, gumbel-softmax/concrete distribution и т.д.). Выполняется составление функции потерь как взвешенной суммы расхождения значений агрегирующей функции и компоненты, отвечающей за минимальность взятой подпоследовательности (например, среднему значению всех бинарных меток). Затем осуществляется процедура градиентного спуска по бинарным меткам для данной последовательности до сходимости.

Во втором примере реализации подпоследовательность (11) выявляется на основании вероятностной рекуррентной модели. Эта модель будет обусловлена по исходной последовательности (10). Под

обуславливанием понимается передача информации об исходной последовательности (10) в вероятностную модель. Для каждого нового токена в исходной последовательности (10) модель будет генерировать случайную величину, определяющую, на основе уже включенных в подпоследовательность (11) токенов, включать ли этот новый токен в подпоследовательность. Реализовать такую схему можно, например, с помощью РНС и concrete distribution для дифференцируемого генерирования.

Мера расхождения может представлять собой заданную функцию или обучаемую модель машинного обучения, например одну или несколько ИНС, которая принимает на вход значения агрегирующих функций.

В случае построения меры расхождения как дискриминатора можно предложить следующую схему его обучения: в качестве положительных примеров на вход дискриминатора будут поступать пары из результата вычисления агрегирующей функции для полной последовательности и для случайной ее аугментации, а в качестве отрицательных - для полной последовательности и найденной подпоследовательности.

Под случайной аугментацией понимается исключение малого количества токенов из исходной последовательности случайным образом (например, независимо разыгрывая бинарную случайную величину для каждого токена). При этом может осуществляться отбор для обучения только тех случайных аугментаций, для которых значение агрегирующей функции мало различается в смысле известных и зафиксированных функций расхождения.

На фиг. 7 представлен пример общего вида вычислительной системы (300), которая обеспечивает реализацию заявленного способа или является частью системы, например сервером, персональным компьютером, частью вычислительного кластера, обрабатывающим необходимые данные для осуществления заявленного технического решения.

В общем случае, система (300) содержит объединенные общей шиной информационного обмена один или несколько процессоров (301), средства памяти, такие как ОЗУ (302) и ПЗУ (303), интерфейсы ввода/вывода (304), устройства ввода/вывода (305), и устройство для сетевого взаимодействия (306).

Процессор (301) (или несколько процессоров, многоядерный процессор и т.п.) может выбираться из ассортимента устройств, широко применяемых в настоящее время, например, таких производителей, как: Intel™, AMD™, Apple™, Samsung Exynos™, MediaTEK™, Qualcomm Snapdragon™ и т.п. Под процессором или одним из используемых процессоров в системе (300) также необходимо учитывать графический процессор, например, GPU NVIDIA или Graphcore, тип которых также является пригодным для полного или частичного выполнения способа, а также может применяться для обучения и применения моделей машинного обучения в различных информационных системах.

ОЗУ (302) представляет собой оперативную память и предназначено для хранения исполняемых процессором (301) машиночитаемых инструкций для выполнения необходимых операций по логической обработке данных. ОЗУ (302), как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.). При этом в качестве ОЗУ (302) может выступать доступный объем памяти графической карты или графического процессора.

ПЗУ (303) представляет собой одно или более устройств постоянного хранения данных, например жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

Для организации работы компонентов системы (300) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (304). Выбор соответствующих интерфейсов зависит от конкретного исполнения вычислительного устройства, которые могут представлять собой, не ограничиваясь: PCI, AGP, PS/2, IrDa, FireWire, LPT, COM, SATA, IDE, Lightning, USB (2.0, 3.0, 3.1, micro, mini, type C), TRS/Audio jack (2.5, 3.5, 6.35), HDMI, DVI, VGA, Display Port, RJ45, RS232 и т.п.

Для обеспечения взаимодействия пользователя с вычислительной системой (100) применяются различные средства (305) В/В информации, например клавиатура, дисплей (монитор), сенсорный дисплей, тач-пад, джойстик, манипулятор, мышь, световое перо, стилус, сенсорная панель, трекбол, динамики, микрофон, средства дополненной реальности, оптические сенсоры, планшет, световые индикаторы, проектор, камера, средства биометрической идентификации (сканер сетчатки глаза, сканер отпечатков пальцев, модуль распознавания голоса) и т.п.

Средство сетевого взаимодействия (306) обеспечивает передачу данных посредством внутренней или внешней вычислительной сети, например Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств (306) может использоваться, но не ограничиваясь: Ethernet карта, GSM модем, GPRS модем, LTE модем, 5G модем, модуль спутниковой связи, NFC модуль, Bluetooth и/или BLE модуль, Wi-Fi модуль и др.

Дополнительно могут применяться также средства спутниковой навигации в составе системы (300), например GPS, ГЛОНАСС, BeiDou, Galileo.

Представленные материалы заявки раскрывают предпочтительные примеры реализации технического решения и не должны трактоваться как ограничивающие иные, частные примеры его воплощения, не выходящие за пределы испрашиваемой правовой охраны, которые являются очевидными для специалистов соответствующей области техники.

Источники информации

1. Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., and Stewart, W. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3504–3512. Curran Associates, Inc., 2016.
2. Choi, E., Bahadori, M. T., Song, L., Stewart, W. F., and Sun, J. Gram: Graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17*, pp. 787–795, New York, NY, USA, 2017. ACM.
3. Ma, F., Chitta, R., Zhou, J., You, Q., Sun, T., and Gao, J. Dipole: Diagnosis prediction in healthcare via attention based bidirectional recurrent neural networks. In *KDD, 2017*.
4. Karpathy, A., Johnson, J., and Li, F. Visualizing and understanding recurrent networks. CoRR, abs/1506.02078, 2015. URL <http://arxiv.org/abs/1506.02078>.
5. Hasani, R. M., Amini, A., Lechner, M., Naser, F., Grosu, R., and Rus, D. Response characterization for auditing cell dynamics in long short-term memory networks. arXiv preprint arXiv:1809.03864, 2018.
6. Strobel, H., Gehrmann, S., Pfister, H., and Rush, A. M. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 24:667–676, 2018.
7. Murdoch, W. J. and Szlam, A. Automatic rule extraction from long short term memory networks. In *International Conference on Learning Representations, 2017*.
8. Murdoch, W. J., Liu, P. J., and Yu, B. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *International Conference on Learning Representations, 2018*. URL <https://openreview.net/forum?id=rkRwGg-0Z>.
9. Yu, K., Liu, Y., Schwing, A. G., and Peng, J. Fast and accurate text classification: Skimming, rereading and early stopping, 2018.
10. Ryabinin, M. and Lobacheva, E. Adaptive prediction time for sequence classification, 2018.
11. Goodfellow et al. Generative Adversarial Nets // Departement d'informatique et de recherche operationnelle / Universite de Montr eal / Montreal, QC H3C 3J7. 2014.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Компьютерно-реализуемый способ отладки обученной рекуррентной нейронной сети (РНС), выполняемый с помощью по меньшей мере одного процессора, причем РНС обучена на наборе данных, состоящем из последовательностей токенов, которые являются векторными представлениями элементов упомянутого набора данных, и способ содержит этапы, на которых:

а) получают значение агрегирующей функции скрытых состояний РНС для упомянутой последовательности токенов;

б) осуществляют поиск внутри упомянутой последовательности токенов по меньшей мере одной подпоследовательности токенов и определяют для каждой упомянутой подпоследовательности агрегирующую функцию скрытых состояний РНС;

с) определяют подпоследовательность токенов на основании минимального значения меры расхождения между значениями агрегирующих функций, полученных на этапах а) и б).

2. Способ по п.1, характеризующийся тем, что мера расхождения является заданной функцией.

3. Способ по п.2, характеризующийся тем, что мера расхождения является обучаемой моделью машинного обучения, представляющей собой по меньшей мере одну искусственную нейронную сеть, принимающую на вход значения агрегирующих функций.

10

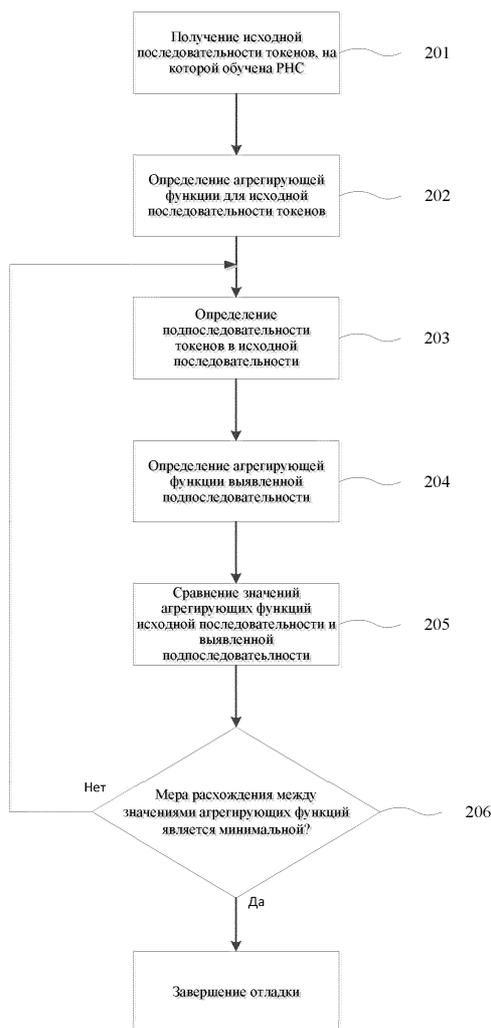
...главным достоинством картины является взор режиссера на людские взаимоотношения. его неповторимого голоса и его неповторимой актёрской игры, но даже в отсутствии него фильм смотрится очень интересно и захватывающе, сейчас трудно представить фильмы без спецэффектов, без пошлого юмора, а ведь больше 40 лет снимали ориентируясь на актёрскую игру и ни на что больше и было просто потрясающе снято, смотреть было одно удовольствие такое кино, где актёры не отбывают номера, а реально живут своими персонажами. Гениальная игра актёров, роль о властвовании в состоянии крайне болезненного истощения, роль о борьбе «ума» и природы, роль о неуживчивости чувства справедливости, о вероятной inferнальности ума, грандиозный фильм своего как для своего, так и для всех времен...

Фиг. 1

11

достоинством
неповторимого неповторимой
очень интересно и захватывающе,
потрясающе снято, смотреть удовольствие
Гениальная
игра актёров,
грандиозный фильм

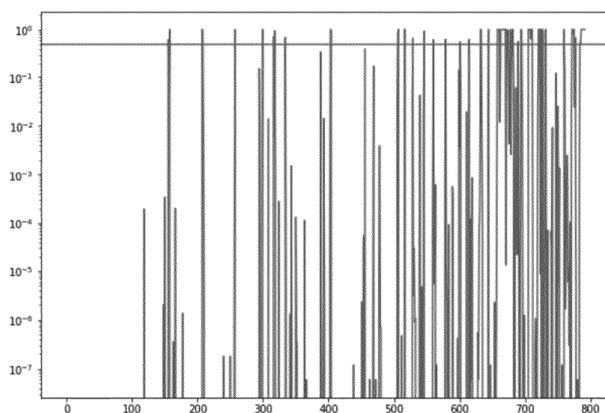
Фиг. 2



Фиг. 3

Значение вероятности РНС = 0.02057

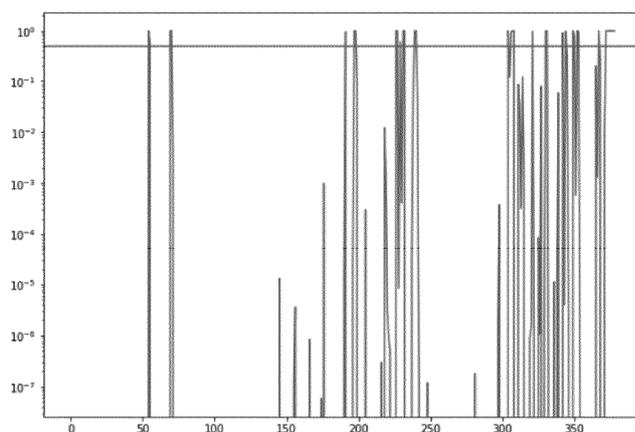
Полученное значение вероятности = 0.1099



Фиг. 4

Значение вероятности РНС = 0.6632

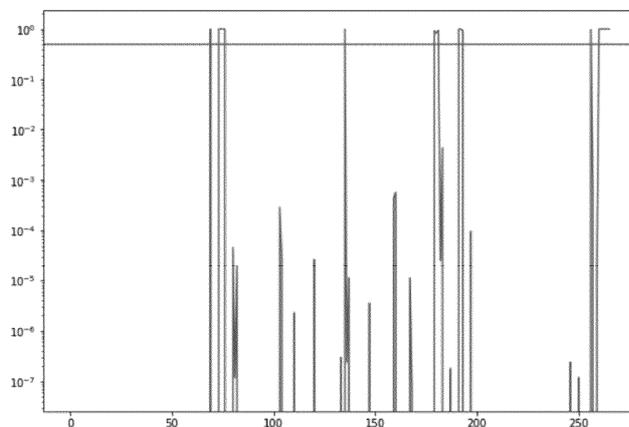
Полученное значение вероятности = 0.7569



Фиг. 5

Значение вероятности РНС = 0.9931

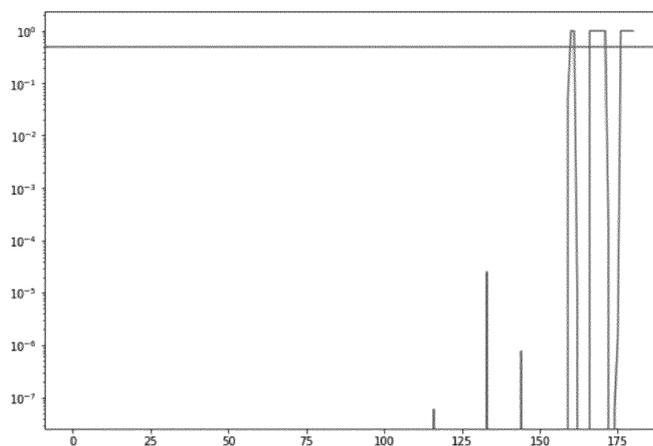
Полученное значение вероятности = 0.9597



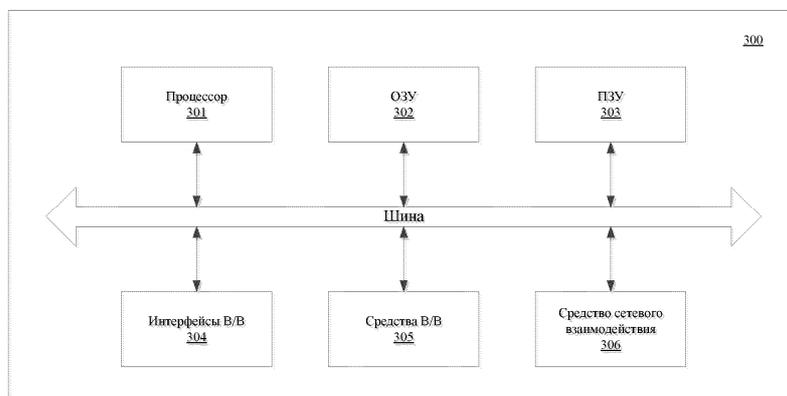
Фиг. 6

Значение вероятности РНС = 0.9903

Полученное значение вероятности = 0.9922



Фиг. 7



Фиг. 8



Евразийская патентная организация, ЕАПВ

Россия, 109012, Москва, Малый Черкасский пер., 2